

3. (Original) The method of claim 1, wherein if the object association is a one to many object association, the step of updating the target value holder includes:
- adding the new target to the target value holder collection from the source, if the collection does not already contain the target;
 - and the step of updating the source value holder includes:
 - removing the source from the old source value holder in the target, if the collection already contains the source; and
 - setting the new source in the source value holder from the target.
4. *115* (Original) The method in claim 1, wherein if the object association is a many to one object association, the step of updating the target value holder includes:
- setting the new target in the target value holder from the source, and
 - removing the target from the old target value holder collection in the source, if the collection already contains the target;
 - and the step of updating the source value holder includes:
 - adding the new source to the source value holder collection from the target, if the collection does not already contain the source.
5. (Original) The method in claim 1, wherein if the object association is a many to many object association, the step of updating the target value holder includes:
- adding the new target to the target value holder collection from the source, if the collection does not already contain the target;
 - and the step of updating the source value holder includes:
 - adding the new source to the source value holder collection from the target, if the collection does not already contain the source.
6. (Original) The method in claim 1, wherein the value holder includes a method for setting source and target objects to be wrapped by the value holder, and a method for returning source and target objects wrapped by the value holder.

7. (Original) The method in claim 1, wherein the steps of updating a target and updating a source are performed using a value holder implemented in Java, C++, Smalltalk, Eiffel, or other object oriented language.

8. (Original) The method in claim 1, wherein the value holder receives as input parameters one or more of:

- AS
- a value holder that an old source currently points to;
 - a value holder that an old target currently points to;
 - a value holder that a source currently points to;
 - a value holder that a target currently points to;
 - a source value;
 - a target value;
 - a source cardinality; and
 - a target cardinality.

9. (Currently amended) A method of deleting an object association between a source object and a target object, comprising:

- updating a source value holder of the source object to identify the target object, the source value holder comprising a value holder that the source object points to; and
 - updating a target value holder of the target object to identify the source object, the target value holder comprising a value holder that the target object points to;
- wherein a value holder is an object that wrappers a target object or source object in order to implement a proxy pattern.

10. (Original) The method of claim 9, wherein if the object association is a one to many object association, the step of updating the source value holder includes:

- setting the value in the source value holder to null;
- and the step of updating the target value holder includes:
 - removing the target from the target value holder collection in the source, if the collection already contains the target.

11. (Original) The method of claim 9, wherein if the object association is a many to many object association, the step of updating the source value holder includes:
removing the source from the source value holder collection in the target, if the collection already contains the source;
and the step of updating the target value holder includes:
removing the target from the target value holder collection in the source, if the collection already contains the target.
12. (Original) The method of claim 9, wherein the value holder includes a method for setting source and target objects to be wrapped by the value holder, and a method for returning source and target objects wrapped by the value holder.
- A5 13. (Original) The method of claim 9, wherein the steps of updating a source and updating a target are performed using a value holder implemented in Java, C++, Smalltalk, Eiffel, or other object oriented language.
14. (Original) The method of claim 9, wherein the value holder receives as input parameters one or more of:
a value holder that a source currently points to;
a value holder that a target currently points to;
a source value;
a target value;
a source cardinality; and
a target cardinality.
15. (Currently amended) A computer program product in a computer readable medium for updating an object association between a source object and a target object, comprising:
first instructions for updating a target value holder of the source target to identify the source object, the target value holder comprising a value holder that the target object points to; and

second instructions for updating a source value holder of the source object to identify the target object, the source value holder comprising a value holder that the source object points to;

wherein a value holder is an object that wrappers a target object or source object in order to implement a proxy pattern.

16. (Original) The computer program product of claim 15, wherein if the object association is a one to one object association, the first instructions for updating the target value holder includes:

setting the new target in the target value holder from the source, and
removing the target from the old target value holder in the source, if the collection already contains the target;

and the second instructions for updating the source value holder includes:

removing the source from the old source value holder in the target, if the collection already contains the source; and

setting the new source in the source value holder from the target.

17. (Original) The computer program product of claim 15, wherein if the object association is a one to many object association, the first instructions for updating the target value holder includes:

adding the new target to the target value holder collection from the source, if the collection does not already contain the target;

and the second instructions for updating the source value holder includes:

removing the source from the old source value holder in the target, if the collection already contains the source; and

setting the new source in the source value holder from the target.

18. (Original) The computer program product in claim 15, wherein if the object association is a many to one object association, the first instructions for updating the target value holder includes:

setting the new target in the target value holder from the source, and

removing the target from the old target value holder collection in the source, if the collection already contains the target;

and the second instructions for updating the source value holder includes:

adding the new source to the source value holder collection from the target, if the collection does not already contain the source.

19. (Original) The computer program product in claim 15, wherein if the object association is a many to many object association, the first instructions for updating the target value holder includes:

adding the new target to the target value holder collection from the source;

and the second instructions for updating the source value holder includes:

adding the new source to the source value holder collection from the target, if the collection does not already contain the source.

20. (Original) The computer program product of claim 15, wherein the value holder includes instructions for setting source and target objects to be wrapped by the value holder, and instructions for returning source and target objects wrapped by the value holder.

21. (Original) The computer program product in claim 15, wherein the first instructions for updating a target and the second instructions for updating a source are performed using a value holder implemented in Java, C++, Smalltalk, Eiffel, or other object oriented language.

22. (Original) The computer program product in claim 15, wherein the value holder receives as input parameters one or more of:

a value holder that an old source currently points to;

a value holder that an old target currently points to;

a value holder that a source currently points to;

a value holder that a target currently points to;

a source value;

a target value;
a source cardinality; and
a target cardinality.

23. (Currently amended) A computer program product in a computer readable medium for deleting an object association between a source object and a target object, comprising:

first instructions for updating a source value holder of the source object to identify the target object, the source value holder comprising a value holder that the source object points to; and

second instructions for updating a target value holder of the target object to identify the source object, the target value holder comprising a value holder that the target object points to;

wherein a value holder is an object that wraps a target object or source object in order to implement a proxy pattern.

24. (Original) The computer program product of claim 23, wherein if the object association is a one to many object association, the first instructions for updating the source value holder includes:

setting the value in the source value holder to null;

and the second instructions for updating the target value holder includes:

removing the target from the target value holder collection in the source, if the collection already contains the target.

25. (Original) The computer program product of claim 23, wherein if the object association is a many to many object association, the first instructions for updating the source value holder includes:

removing the source from the source value holder collection in the target, if the collection already contains the source;

and the second instructions for updating the target value holder includes:

removing the target from the target value holder collection in the source, if the collection already contains the target.

26. (Original) The computer program product of claim 23, wherein the value holder includes instructions for setting source and target objects to be wrapped by the value holder, and instructions for returning source and target objects wrapped by the value holder.

27. (Original) The computer program product of claim 23, wherein the first instructions for updating an old source and the second instructions for updating a target are performed using a value holder implemented in Java, C++, Smalltalk, Eiffel, or other object oriented language.

A5
28. (Original) The computer program product of claim 23, wherein the value holder receives as input parameters one or more of:

- a value holder that a source currently points to;
- a value holder that a target currently points to;
- a source value;
- a target value;
- a source cardinality; and
- a target cardinality.

29. (Currently amended) An apparatus for updating an object association between a source object and a target object, comprising:

means for updating a target value holder of the target object to identify the source object, the target value holder comprising a value holder that the target object points to;
and

means for updating a source value holder of the source object to identify the target object, the source value holder comprising a value holder that the source object points to;

wherein a value holder is an object that wrappers a target object or source object in order to implement a proxy pattern.

30. (Currently amended) An apparatus for deleting an object association between a source object and a target object, comprising:

means for updating a source value holder of the source object to identify the target object, the source value holder comprising a value holder that the source object points to;
and

means for updating a target value holder of the target object to identify the source object, the target value holder comprising a value holder that the target object points to;

wherein a value holder is an object that wrappers a target object or source object in order to implement a proxy pattern.
